

Method and Apparatus For Capturing And Rendering

Annotations For Non-Modifiable Electronic Content

1. Related Applications

This application is related to the following applications filed herewith:

5 U.S. Serial No. (BW 03797.80027), filed December , 1999, entitled "Bookmarking and Placemarking a Displayed Document in a Computer System;"

U.S. Serial No. (BW 03797.84619), filed December , 1999, entitled "Method and Apparatus for Installing and Using Reference Materials In Conjunction With Reading Electronic Content;"

10 U.S. Serial No. (BW 03797.84618), filed December , 1999, entitled "Method and Apparatus for Capturing and Rendering Text Annotations For Non-Modifiable Electronic Content;"

U.S. Serial No. (BW 03797.84809) filed , entitled "System, Method, and User Interface for Active Reading of Electronic Content;" and

15 U.S. Serial No. (BW 03797.78802), filed December , 1999, entitled "System and Method for Annotating an Electronic Document Independently of Its Content."

2. Background

Technical Field

The disclosure generally relates to the electronic display of documents. More particularly, 20 the disclosure relates to the placement and use of annotations in electronically displayed documents.

Related Art

Many factors today drive the development of computers and computer software. One of these factors is the desire to provide accessibility to information virtually anytime and anywhere. The proliferation of notebook computers, personal digital assistants (PDAs), and other personal electronic devices reflect the fact that users want to be able to access information wherever they may be, whenever they want. In order to facilitate greater levels of information accessibility, the presentation of information must be made as familiar and comfortable as possible.

In this vein, one way to foster success of electronic presentations of information will be to allow users to handle information in a familiar manner. Stated another way, the use and manipulation of electronically-presented information may mimic those paradigms that users are most familiar with, e.g., printed documents, as an initial invitation to their use. As a result, greater familiarity between users and their “machines” will be engendered, thereby fostering greater accessibility, even if the machines have greater capabilities and provide more content to the user beyond the user’s expectations. Once users feel comfortable with new electronic presentations, they will be more likely to take advantage of an entire spectrum of available functionality.

One manner of encouraging familiarity is to present information in an electronic book format in which a computer displays information in a manner that closely resembles printed books. In order to more completely mimic a printed book, users will need to have the ability to make textual notes to themselves, akin to writing in the margins of paper books. Users will also want to highlight selected portions, as these are active-reading activities of which a user would expect to see in an electronic book. Users will want to add drawings, arrows, underlining, strike-

throughs, and the like, also akin to writing in paper books. Finally, users will want to add bookmarks.

The above-identified so-called “active-reading” activities are available. However, all of these active-reading activities require modification of the underlying document. For example, as 5 is known in the art, if one adds a comment or annotation in an electronic editor, the comment or annotation is inserted into the document. This insertion corrupts the underlying document from its pre-insertion, pristine state. While this may not be an issue in an editable document, the modification of a copyrighted document may run afoul of various copyright provisions. The violations may be compounded with the forwarding of the document to another in its modified 10 state. Further, irrespective of any copyright transgressions, publishing houses responsible for the distribution of the underlying text may not be pleased with any ability to modify their distributed and copyrighted works.

Thus, the users’ desire to actively read and annotate works clashes with the goals of publishing houses to keep copyrighted works in their unmodified state. Without solution of this 15 dilemma, the growth of the electronic publishing industry may be hampered, on one hand, by readers who refuse to purchase electronic books because of the inability to annotate read-only documents and, on the other hand, by the publishing industry that refuses to publish titles that allow for annotations that destroy the pristine compilation of the electronic works.

3. Summary

20 The present invention provides a technique for annotating an electronic document without corruption of the document itself. In the context of the present invention, a “document” encompasses all forms of electronically displayable information including but not limited to books, manuals, reference materials, picture books, etc. To create an annotation, a user selects an

object in the document to locate where the annotation is to be placed. The computer system determines which object has been selected and determines a file position associated with the selected object. The user adds the annotation and, eventually, returns to reading the document. The annotations may be filtered, navigated, sorted, and indexed per user input. Annotations may 5 include text annotations, drawings, highlights, bookmarks, and the like as is related to the general field of active reading.

In the context of the present invention, a displayed “object” may include text, graphics, equations, and other related elements as contained in the displayed document. Annotations may include highlighting, adding textual notes, adding drawings (as one would expect to do with a 10 pencil or pen to a paper book), and adding bookmarks. While the annotations are to be displayed in conjunction with the document, the underlying document is not modified. Related annotations are described in the following disclosures:

U.S. Serial No. (BW 03797.80027), filed December , 1999, entitled “Bookmarking and Placemarking a Displayed Document in a Computer System;”

15 U.S. Serial No. (BW 03797.84619), filed December , 1999, entitled “Method and Apparatus for Installing and Using Reference Materials In Conjunction With Reading Electronic Content;”

20 U.S. Serial No. (BW 03797.84618), filed December , 1999, entitled “Method and Apparatus for Capturing and Rendering Text Annotations For Non-Modifiable Electronic Content;”

U.S. Serial No. (BW 03797.84809) filed , entitled “System, Method, and User Interface for Active Reading of Electronic Content;” and

U.S. Serial No. (BW 03797.78802), filed December , 1999, entitled "System and Method for Annotating an Electronic Document Independently of Its Content,"

which are incorporated herein by reference in their entireties for any enabling disclosure.

To associate an annotation with a selected object, the annotations are linked to a file
5 position in the non-modifiable document. The invention calculates the file position of, for example, the first character of the word (or other displayed element) and stores the file position with the annotation in a separate, linked local file. Alternatively, the non-modifiable document may represent a non-modifiable portion of a file, with the annotations being added to a write-enabled portion of the file.

10 The determined file position may be used for direct random access into the non-modifiable document despite the document being compressed or decompressed. In one embodiment, the file position is specified in a UTF-8 (a known textual storage format) document derived from an original Unicode (another known textual storage format). However, in order to conserve space, the non-modifiable document may be compressed using a general-purpose
15 binary compression algorithm, decompressed, and translated to Unicode for viewing. Accordingly, the file position as stored for an annotation is consistent through various storage schemes and compression techniques.

These and other novel advantages, details, embodiments, features and objects of the present invention will be apparent to those skilled in the art from following the detailed
20 description of the invention, the attached claims and accompanying drawings, listed herein, which are useful in explaining the invention.

4. Brief Description of Drawings

Figure 1 shows a general purpose computer supporting the display and annotation of an electronic document in accordance with embodiments of the present invention.

Figure 2 shows a displayed document on a computer screen in accordance with
5 embodiments of the present invention.

Figures 3A and 3B show different document formats available for storing a document in accordance with embodiments of the present invention.

Figure 4 shows different bytes for storing characters in UTF8 and Unicode in accordance with embodiments of the present invention.

10 Figure 5 shows a process for determining the file position of an object in accordance with
embodiments of the present invention.

Figure 6 shows another process for determining the file position of an object in accordance with embodiments of the present invention.

15 Figure 7 shows a process for displaying annotations in accordance with embodiments of
the present invention.

Figures 8A and 8B show various storage techniques for storing annotations in accordance with embodiments of the present invention.

Figure 9 shows a screen for manipulating annotations in accordance with embodiments of
the present invention.

20 5. Detailed Description

The present invention relates to a system and method for capturing annotations associated with a non-modifiable document.

Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules. Generally, program modules include routines, programs, objects, scripts, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will

5 appreciate that the invention may be practiced with any number of computer system configurations including, but not limited to, distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices. The present invention may also be practiced in personal computers

10 (PCs), hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like.

Figure 1 is a schematic diagram of a computing environment in which the present invention may be implemented. The present invention may be implemented within a general purpose computing device in the form of a conventional personal computer 200, including a

15 processing unit 210, a system memory 220, and a system bus 230 that couples various system components including the system memory to the processing unit 210. The system bus 230 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 240 and random access memory (RAM) 250.

20 A basic input/output system 260 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 200, such as during start-up, is stored in ROM 240. The personal computer 200 further includes a hard disk drive 270 for reading from and writing to a hard disk, not shown, a magnetic disk drive 280 for reading from

or writing to a removable magnetic disk 290, and an optical disk drive 291 for reading from or writing to a removable optical disk 292 such as a CD ROM or other optical media. The hard disk drive 270, magnetic disk drive 280, and optical disk drive 291 are connected to the system bus 230 by a hard disk drive interface 292, a magnetic disk drive interface 293, and an optical disk drive interface 294, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 200.

Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 290 and a removable optical disk 292, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 290, optical disk 292, ROM 240 or RAM 250, including an operating system 295, one or more application programs 296, other program modules 297, and program data 298. A user may enter commands and information into the personal computer 200 through input devices such as a keyboard 201 and pointing device 202. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 210 through a serial port interface 206 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 207 or other type of display device is also connected to the system bus

230 via an interface, such as a video adapter 208. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The personal computer 200 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 209. The remote computer 209 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 200, although only a memory storage device 211 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 212 and a wide area network (WAN) 213. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 200 is connected to the local network 212 through a network interface or adapter 214. When used in a WAN networking environment, the personal computer 200 typically includes a modem 215 or other means for establishing a communications over the wide area network 213, such as the Internet.

The modem 215, which may be internal or external, is connected to the system bus 230 via the serial port interface 206. In a networked environment, program modules depicted relative to the personal computer 200, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In addition to the system described in relation to Figure 1, the invention may be practiced on a handheld computer. Further, purpose-built devices may support the invention as well. In short, handheld computers and purpose-built devices are similar in structure to the system of Figure 1 but may be limited to a display (which may be touch-sensitive to a human finger or

stylus), memory (including RAM and ROM), and a synchronization/modem port for connecting the handheld computer and purpose-built devices to another computer or a network (including the Internet) to download and/or upload documents or download and/or upload annotations. The description of handheld computers and purpose-built devices is known in the art and is omitted for simplicity. The invention may be practiced using C. Also, it is appreciated that other languages may be used including C++, assembly language, and the like.

Figure 2 shows a displayed document on a computer screen in accordance with embodiments of the present invention. As preferred, the document is displayed in a form that closely resembles the appearance of a paper equivalent of the e-book and, in this case, a paper novel. The document reader window 101 may comprise a variety of portions including a title bar 101 listing the title of the document and a body 102. In the body 102 of the display window, various portions of a document may be displayed. Figure 2 shows an example where a title 104, a chapter number 105, a chapter title 106, and the text of the chapter 107 are displayed. Similar to an actual book, margins 108, 109, 110, and 111 appear around the displayed text. As referred to herein, the displayed elements may be independently referenced. Here for example, object 103 “we” has a drawing annotation placing a box around it as placed there by the user.

Various schemes exist with which to store electronically displayable information as shown in Figures 3A and 3B. With respect to the storage of text, the industry standard is Unicode UCS-2. Unicode UCS-2 encodes text using two bytes per character. The letters from the standard English alphabet to complex symbols including foreign letters and symbols are all encoded using two bytes. Figure 3A shows a pictorial representation of a four-letter word as stored in four pairs of bytes. Another storage scheme includes UTF-8 in which standard letters (for example, US-ASCII characters) are encoded using only a single byte. Foreign characters and symbols from the

Unicode UCS-2 set are encoded with two or three bytes. Part of the first byte is used to indicate how many total bytes define the complete character as shown in Figure 3B. The remaining bytes are restricted to numeric values that cannot be confused with those used to define a single-byte character. For large texts using standard letters, a UTF8-encoded file may have a size half of that
5 as Unicode. However, in the situation in which a number of foreign characters or symbols, the size of the stored file may actually be larger than that of Unicode due to the greater number of three byte representations of a letter or symbol. Other variable byte-length character encodings have been used in industry, for example, the Shift-JIS standard encodes characters (drawn from a smaller set than Unicode draws from) in one or two bytes. Unlike in UTF-8, the second byte of a
10 two-byte character may contain a value that may also be used by itself to represent a single-byte character.

Figure 4 shows different bytes for storing characters in UTF8 and Unicode in accordance with embodiments of the present invention. An example of the two schemes discussed with respect to Figures 3A and 3B is shown in Figure 4. The word “banana” takes twelve bytes to represent it in Unicode while only using six bytes in UTF8. The word “façade” requires twelve bytes in Unicode and seven bytes in UTF8. Other storage schemes are known in the art but not shown here for simplicity. The difference between UTF8 and Unicode is provided by way of example only and not intended to limit the invention to the use of storage scheme over the other.
15

The difference in the storage modes becomes relevant in the technique used to fix the file
20 position for an annotation. If the file position is determined with one storage scheme, porting the file position to another storage scheme may not result in the same desired file position for an annotation. Thus, all annotations may be fixed to a file position based on the use of a single scheme. Preferably, the scheme used to hold the document while the document is being

displayed is the scheme that is used to determine the file position. So, irrespective of whether the document is closed and compressed to another scheme, when reopened in the display scheme, the file position for the annotation remains the same as when created. Unicode may be the scheme used to display the document. Alternatively, UTF8 may be used as well as any other 5 textual encoding or compression scheme to access the document for display.

Figure 5 shows a process for determining the file position of an object in accordance with embodiments of the present invention. In step 501, a user selects an object on the screen. The user may select the object via a cursor controlled through a mouse, touch-pad, trackball, or like pointing device. Alternatively, the user may use a stylus or finger if the surface of the display can 10 accommodate such input.

In step 502, the system determines which object was selected by the user. This step relates to the conversion of the physical coordinates from the display device to coordinates inside the reader window. From this conversion, the object selected by the user is known.

Step 502A is optional. It relates to the user selection of an action post selection of the 15 object. If the user is supplied with a menu after selection of the object and the function of adding an annotation is provided on the menu, step 502A relates to the selection of the adding the annotation function. An example of adding an annotation is described in detail in U.S. Serial No. (BW 03797.84618), filed December , 1999, entitled “Method and Apparatus for Capturing and Rendering Text Annotations For Non-Modifiable Electronic Content” whose contents are 20 incorporated by reference for any essential subject matter.

Step 503 relates to the determination of the file position of the selected object. The file position may include the first byte of the selected object. Alternatively, the file position may be the first byte of the last character (or even the character following the last character) of the

selected object. Selecting the first byte of the first character to determine the file position provides the advantage of displaying any annotation on the page of the beginning of the object, rather than on the next page if the object spans a page. Anyone of skill in the art will appreciate that any byte of the selected object (or surrounding the selected object) may be selected to 5 provide the file position of the object. Alternatively, one may select a line in which the object resides or the paragraph or the portion of the page (e.g., the top, middle or bottom of the page).

The file position may be determined by counting the number of bytes from some known file position to the location of, for example, the first character of the selected object. The known file position may be the beginning of the file, or may be, for example, a previously noted file 10 position for the beginning of the current paragraph. The counting step may be performed before or after generation of the annotation. Alternatively, the counting step may be performed in the background while the annotation is being created by the user. Note that annotation file positions may always stored as UTF-8 offsets within the text, as it stood before binary compression. However, the algorithm used to display the text works with Unicode characters. Therefore, in 15 this example, it is necessary to work back from the selected object to a character with a known UTF-8 file position.

Because the binary file-format of the original publication (electronic book, document, etc.) intermixes markup (tags) with text, it is necessary to discount the bytes taken by such tags when calculating the file-position for the selected object (to which the annotation will be 20 anchored). However, of the said tags, many if not most do not take up a character-position on the display surface. Therefore, it is necessary to keep track of the starting file position of every run of text on the display, which corresponds to an unbroken run of text in the file. An “unbroken” run of text refers to text in the file that is not broken by a start- or an end- tag.

Therefore, the steps involved in accurately determining the file position for anchoring the annotation to the selected object are:

- 1) Look up in our data structures what display character position is the start of an “unbroken” run described in the preceding paragraphs.
- 5 2) Fetch from the same data structure the file-position associated with the starting display-character position.
- 3) Determine the string which runs from the run-start position to the selection-start position. This string contains some number of Unicode characters.
- 10 4) Determine how many UTF-8 bytes would be required to hold a UTF-8-encoded version of the string from step 3).
- 5) Add the UTF-8 bytecount from step 4 to the file-position from step 2.

Step 504 relates to creating a file to persist the annotation. While shown after step 503, it will be appreciated that it may occur prior to or during the determination of the file position of the object. In step 505, the file position is placed in the header of the file (or portion of the file) 15 storing the created annotation. Alternatively, the file position may be appended to the file being viewed.

Figure 6 shows another process for determining the file position of an object in accordance with embodiments of the present invention. As shown in step 601, a user navigates to a page. Once on the page, the system determines the file position of the first byte of the first 20 object on the page as shown in step 602. The file position may be determined every time a new page is displayed. Alternatively, the system may pause (for example, two seconds) before starting to determine the file position for the first byte in order to allow the user to navigate to a new page before starting the file position determination. This delay provides the advantage of

minimizing system workload when a user is quickly flipping pages. Once the user settles down with a given page, the system may then determine the file position of the first byte.

In step 603, the file position of the page is temporarily stored in memory.

In step 604, the system waits for either selection of an object or navigation to another 5 page. More options are contemplated that do not need the file position for execution (for example, looking up a term in a reference document as disclosed in U.S. Serial No. (BW 03797.84619) filed , entitled “Method and Apparatus for Installing and Using Reference Materials In Conjunction With Reading Electronic Content”, whose contents are incorporated herein by reference in its entirety for any enabling disclosure).

10 In step 605, once an object is selected, the relative position of the selected object is determined with reference to the first byte of the first object on the displayed page.

15 In step 606, the file position of the first byte of the first object on the page as determined in step 602 is retrieved from memory (as stored in step 603) and added to the relative position of the first byte of the selected object as determined in step 605 to determine the file position of the selected object.

20 In step 607, the file position of the selected object is stored along with the created annotation. These steps relating to the determination of the file position may occur before or after the annotation for the object. Alternatively, the file position may be preformed in the background while the annotation is being created. Those skilled in the art will appreciate that any number of techniques may be used to determine object position and still be considered to be within the scope of the present invention.

Figure 7 relates to a process for displaying the created annotation when navigating to the page. In step 701, a user navigates to a page.

In step 702, the system determines the file position of the first object on the page.

In step 703, the system determines the file position of the last object on the page.

In step 704, the annotations stored for the document are searched to determine if any have file positions located between the file position determined in step 702 and the file position
5 determined in step 703.

In step 705, if no annotations with a file position are located for display on the displayed page, the system waits for user input (including, for example, navigation to a new page or selection of an object for annotation, or any other action described herein).

In step 706, an annotation has been found that relates to an object on the page. The
10 location of the object on the page is determined and the annotation is displayed for the object. The system for determining the location of the object may include subtracting the file position of the first object on the page from the file position of the annotated object. This difference is then used to determine how many bytes from the first character of the page is the annotated object. At this point, further annotations may be made, by returning from step 706 to step 705.

15 Alternatively, the system may count again from the beginning of the document to determine which object has been annotated. It will be appreciated by those skilled in the art that numerous methods exist for displaying the annotation for the annotated object. The above examples are not intended to be limiting.

In the context of displaying the annotations that are determined to exist in a given “page”
20 of the content (the unit of text being viewed by the user at any given time), the computer system will first validate a global state, which determines whether annotations should be rendered at all. For example, the user is provided with the ability to globally specify whether to show or hide drawing annotations (as well as text notes, bookmarks, highlights, etc.). Prior to displaying a

particular annotation of an object, the computer system will check this global setting to determine whether or not to render the specific annotation. If the user has chosen to hide annotations of that particular type, the annotation will not be rendered.

Figures 8A and 8B show various storage techniques for storing annotations in accordance with embodiments of the present invention. Figure 8A shows a document 801 that has modifiable (803-806) and non-modifiable (802) portions. Files of this type include Infotext file formats as are known in the art. Annotations 806 may be stored in combination with the non-modifiable content 802. An annotation 806 may be stored in a file with header 803 and body 806. The header 803 includes, for example, the file position 804 of the object with which the annotation 806 is associated. It may also include an indication of the type of annotation 806 in file portion 805. As discussed above, the annotation 806 may include a highlight, a bookmark, a drawing to be overlaid over the object, or a text annotation.

Figure 8B shows the non-modifiable content 809 as a separate file apart from the annotation file. The annotation file 807 of Figure 8B has similar constituent elements to that of annotation 807 of Figure 8A. Annotation file 807 may include a file portion 808 that indicates to which non-modifiable document (here, 809) it is linked. Using the approach set forth in Figure 8B, one file may store all annotations for a user with the non-modifiable content portions 809 being stored separately. This approach has the advantage of being able to quickly scan all annotations at one time rather than accessing all documents 801 (as including non-modifiable portions 802) to obtain all annotations stored therein.

Figure 9 shows a display window for sorting, modifying, searching, and renaming the annotations stored in a system. The window 900 includes a title identifier 901 to alert the user that he or she is in an annotation pane 900. The window 900 may include two panes 902 and 903

(other panes may be added as needed). Panes 902 and 903 may provide a listing of annotations 904 by document. Alternatively, they may provide a listing of all annotations in a person's system. When in pane 902 (here, entitled "Notes"), the user may sort the list of annotations by type (highlight, drawing, text, bookmark). Selecting an annotation allows one to navigate to the 5 location in the document containing the annotation. Selecting and holding the annotation allows one to remove, change the appearance of, hide or show that particular annotation, or rename the annotation. The second pane 903 (here, entitled "View") may allow a user to sort annotations based on their properties. For example, one may sort on the time created, time last accessed, by 10 type, alphabetically, and on book order. Further, individual annotations may be switched on or off using controls on the page. Also, if all annotations have been switched off (or just those of a specific type of annotations have been switched off) and another annotation is created (or another annotation in that class), all annotations of that type may be switched back on. This may be extended to include all annotations being switched on if hidden and a new annotation added.

In the foregoing specification, the present invention has been described with reference to 15 specific exemplary embodiments thereof. Although the invention has been described in terms of various embodiments, those skilled in the art will recognize that various modifications, embodiments or variations of the invention can be practiced within the spirit and scope of the invention as set forth in the appended claims. All are considered within the sphere, spirit, and scope of the invention. The specification and drawings are, therefore, to be regarded in an 20 illustrative rather than restrictive sense. Accordingly, it is not intended that the invention be limited except as may be necessary in view of the appended claims.